

1)Create Network Load Balancer.

Step I : Create two instances in EC2 using below scripts in advanced settings.

```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "this is ROOT $HOSTNAME" > /var/www/html/index.html
```

```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "this is my HOME $HOSTNAME" > /var/www/html/index.html
```

Instances (2) [Info](#)

Find Instance by attribute or tag (case-sensitive) Any state

<input type="checkbox"/>	Name ↗ ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS ▼	Public IPv4 ... ▼	Elast...
<input type="checkbox"/>	ninst1	i-06a32d4b385306518	✔ Running 🔍 🔍	t2.micro	🔄 Initializing	View alarms +	us-east-1c	ec2-18-208-231-156.co...	18.208.231.156	-
<input type="checkbox"/>	ninst2	i-07d87c9538a374819	⌚ Pending 🔍 🔍	t2.micro	-	View alarms +	us-east-1c	ec2-18-234-114-245.co...	18.234.114.245	-

Step II : In Load Balancing, click load Balancer and then click the create Load Balancer.

[EC2](#) > Load balancers

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<div>No load balancers</div> <div>You don't have any load balancers in us-east-1</div> <div>Create load balancer</div>							

Step III : then Select the network Load Balancer.

EC2 > Load balancers > Compare and select load balancer type

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

Classic Load Balancer - previous generation

Close

Step IV : give the name,select scheme,Ip address type, select all zones .

EC2 > Load balancers > Create Network Load Balancer

Create Network Load Balancer

The Network Load Balancer distributes incoming TCP and UDP traffic across multiple targets such as Amazon EC2 instances, microservices, and containers. When the load balancer receives a connection request, it selects a target based on the protocol and port that are specified in the listener configuration, and the routing rule specified as the default action.

► How Network Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

ntlb

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme
Scheme can't be changed after the load balancer is created.

☒ Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type
Select the type of IP addresses that your subnets use.

☒ IPv4
Recommended for internal load balancers.

☐ Dualstack
Includes IPv4 and IPv6 addresses.

Network mapping

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

vpc-0ee2a03913c112cf
IPv4: 172.31.0.0/16

Mappings
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC can't be selected. Subnets can be added, but not removed, once a load balancer is created.

CloudShell Feedback

Select at least one availability zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC can't be selected. Subnets can be added, but not removed, once a load balancer is created.

☒ **us-east-1a (use1-az2)**
 Subnet: subnet-09b643a39c0fa9c5e
 IPv4 address: Assigned by AWS

☒ **us-east-1b (use1-az4)**
 Subnet: subnet-011bedca2c5808a49
 IPv4 address: Assigned by AWS

☒ **us-east-1c (use1-az6)**
 Subnet: subnet-07912f3cf02a80344
 IPv4 address: Assigned by AWS

☒ **us-east-1d (use1-az1)**
 Subnet: subnet-0b449152f57dc1844
 IPv4 address: Assigned by AWS

☒ **us-east-1e (use1-az3)**
 Subnet: subnet-02311cabb18650214
 IPv4 address: Assigned by AWS

☒ **us-east-1f (use1-az5)**
 Subnet: subnet-066546a950f5789ca
 IPv4 address: Assigned by AWS

Step V : select security group, here in default action click to create target group. create the target group add the instances which we created before and registered it .Add it in Default Action.

Security groups - recommended
 Security groups support on Network Load Balancers can only be enabled at creation by including at least one security group. You can change security groups after creation. The security groups for your load balancer must allow traffic to communicate with registered targets on both the listener port and the health check port. For Amazon ElastiCache, security groups also need to allow all inbound traffic. However, you can turn off inbound rule evaluation after creation within the load balancer's security tab or using this link.

Select up to 5 security groups:

Listeners and routing [info](#)
 A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener: TCP:80

Protocol: TCP Port: 80 Subnets

Default action: **info**
 Forward to: **target** target type: instance, ip
[Create target group](#)

Listener tags - optional
 Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

 You can add up to 50 more tags.

Load balancer tags - optional
 Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The "Key" is required, but "Value" is optional. For example, you can have Key = production/development, or Key = webserver, and Value = production.

Optimize with service integrations - optional
 Optimize your load balancing architecture by integrating AWS services with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the load balancer's "Integrations" tab.

AWS Global Accelerator [info](#)
 Optimizes Performance, Availability, Security [Additional charges apply](#)

[Feedback](#)

EC2 > Target groups > ntgrp > Register targets

Register targets

Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. Once you are satisfied with your selections, click Register pending targets.

Available instances (2)

Filter instances

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
<input type="checkbox"/>	i-07d815953ba274819	ntinst2	Running	launch-wizard-2	us-east-1c	172.31.33.28	subnet-07912fc0f02c803f4	March 6, 2024, 11:57 (UTC+05:30)
<input type="checkbox"/>	i-06a32486385306518	ntinst1	Running	launch-wizard-2	us-east-1c	172.31.39.201	subnet-07912fc0f02c803f4	March 6, 2024, 11:54 (UTC+05:30)

0 selected

Ports for the selected instances

Ports for routing traffic to the selected instances

+80555 Separate multiple ports with commas

Include as pending below

2 selections are now pending below. Include more or register targets when ready.

Review targets

Targets (2)

Filter targets

Show only pending

Remove all pending

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-07d815953ba274819	ntinst2	80	Running	launch-wizard-2	us-east-1c	172.31.33.28	subnet-07912fc0f02c803f4	March 6, 2024, 11:57 (UTC+05:30)
i-06a32486385306518	ntinst1	80	Running	launch-wizard-2	us-east-1c	172.31.39.201	subnet-07912fc0f02c803f4	March 6, 2024, 11:54 (UTC+05:30)

2 pending

Cancel Register pending targets

Step VI : review the summary and click on create Load Balancer.

Create an accelerator

An accelerator will be created in your account. The accelerator provides 2 global static IPs that act as a fixed entry point to your load balancer.

Review

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose Create load balancer.

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)

ntlb

- Internet-facing
- IPv4

Security groups [Edit](#)

- launch-wizard-2
 - [sg-090165aa7827d03d3](#)

Network mapping [Edit](#)

VPC [vpc-0ee2a53913c5112cf](#)

- us-east-1a
 - [subnet-09db653a59e0fa9c5e](#)
- us-east-1b
 - [subnet-011bedea2c5808a49](#)
- us-east-1c
 - [subnet-07912fc0f02c803f4](#)
- us-east-1d
 - [subnet-0b449152f37dc1844](#)
- us-east-1e
 - [subnet-02311cab18650214](#)
- us-east-1f
 - [subnet-066546a950f3789ea](#)

Listeners and routing [Edit](#)

- TCP:80 defaults to [ntgrp](#)

Service integrations [Edit](#)

AWS Global Accelerator: None

Tags [Edit](#)

None

Attributes

Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Creation workflow and status

Server-side tasks and status

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

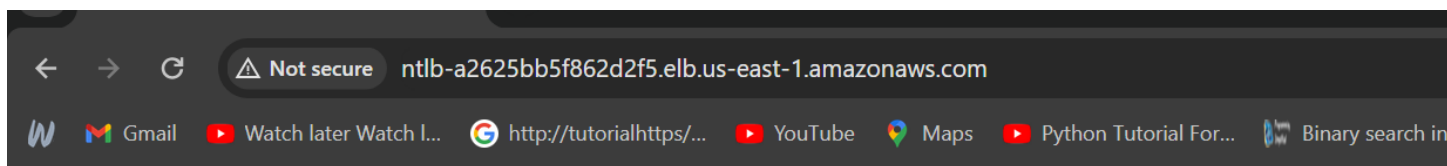
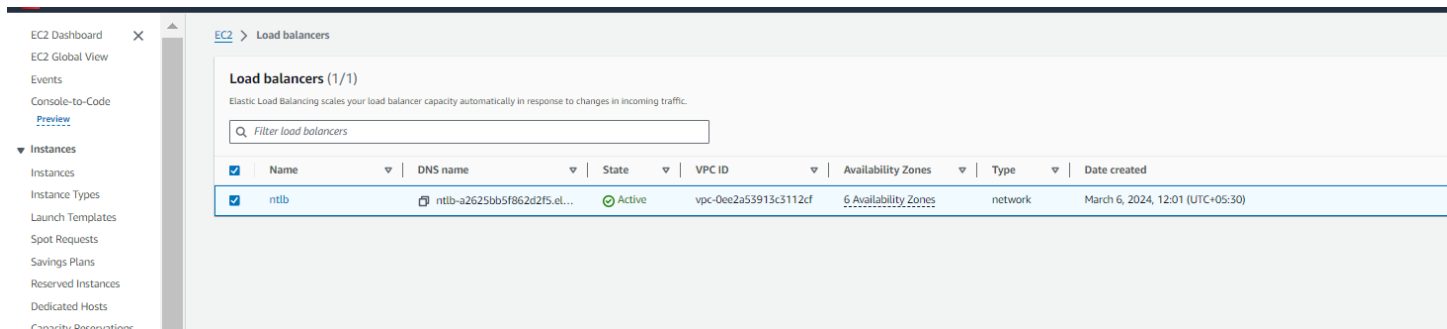
Cancel

Create load balancer

CloudShell

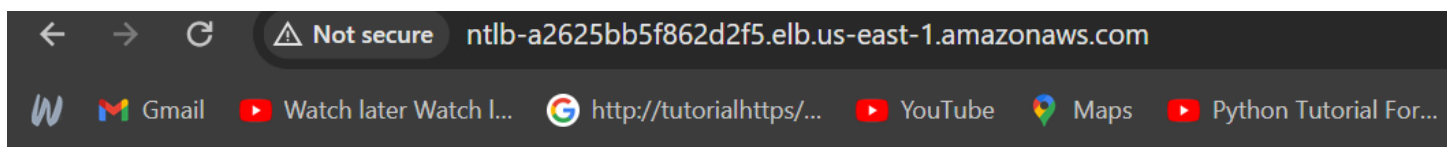
Feedback

Step VII : here, load balancer is created. Copied the DNS Name and paste in other browser.



this is ROOT ip-172-31-33-28.ec2.internal

Step VIII : network load balancer directs the traffic unevenly on reload, it not used round-robin method IP get changes only at the high request occurs.



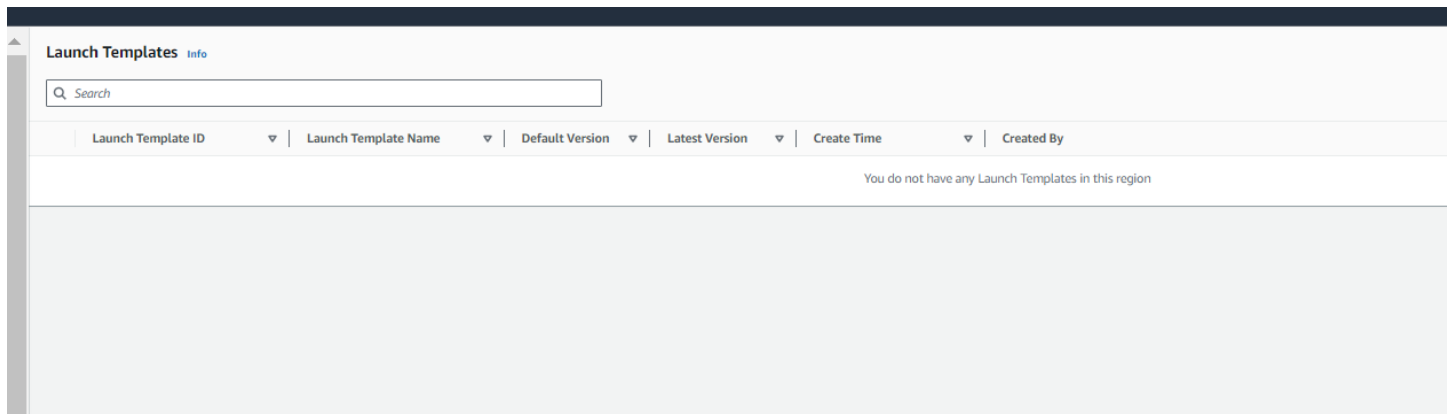
this is my HOME ip-172-31-39-201.ec2.internal

2)AUTO SCALING

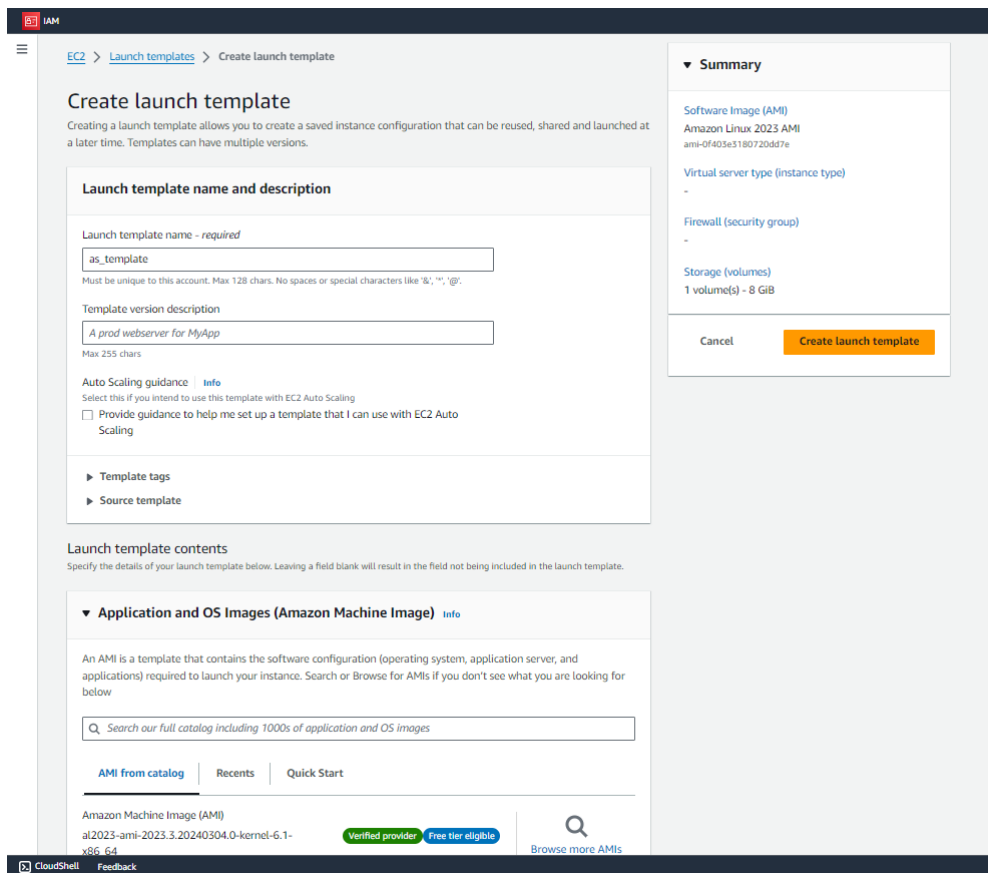
It helps you automatically adjust the number of Ec2 instances in your application based on demands.

1)Create the Template.

Step I : go to Ec2 ,In Instances click on Launch Template then click on create launch template.



Step II : Give the name to template, select AMI and instance type then select existing group which include SSH and HTTP enable.



Instance type [info](#) [for choice](#) Advanced

Instance type Free tier eligible All generations Compare instance types

t2.micro
Family: t2 | vCPU: 1 vCPU | 1 GB Memory | Current generation: t2.micro
On-Demand Linux base pricing: \$0.0125 USD per Hour
On-Demand Linux base pricing: \$0.0125 USD per Hour
On-Demand Linux base pricing: \$0.0125 USD per Hour
Additional costs apply for APIs with pre-installed software

Key pair (login) [info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name: newdata Create new key pair

Network settings [info](#)

Subnet [info](#)
Don't include in launch template Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security group) [info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Select existing security group ☐ Create security group

Security groups [info](#)
Select security group: launch-wizard-2 sg-096165a67827d05d5 Compare security group rules

Advanced network configuration

Storage (volumes) [info](#)

Summary

Software Image (AMI)
Amazon Linux 2023 AMI
ami-0f6c34110712d6b74

Virtual server type (instance type)
t2.micro

Firewall (security group)
launch-wizard-2

Storage (volumes)
1 volume(s) - 8 GB

Cancel Create Launch Template

Step III : In Advanced details, write above script and then click on launch Template.

User data [optional](#) [info](#)
Optional: If you use user data, enter it in the field.

[Choose file](#)

```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "This is Amazon S3 Hosting Template" > /var/www/html/index.html
```

Summary

Software Image (AMI)
Amazon Linux 2023 AMI
ami-0f6c34110712d6b74

Virtual server type (instance type)
t2.micro

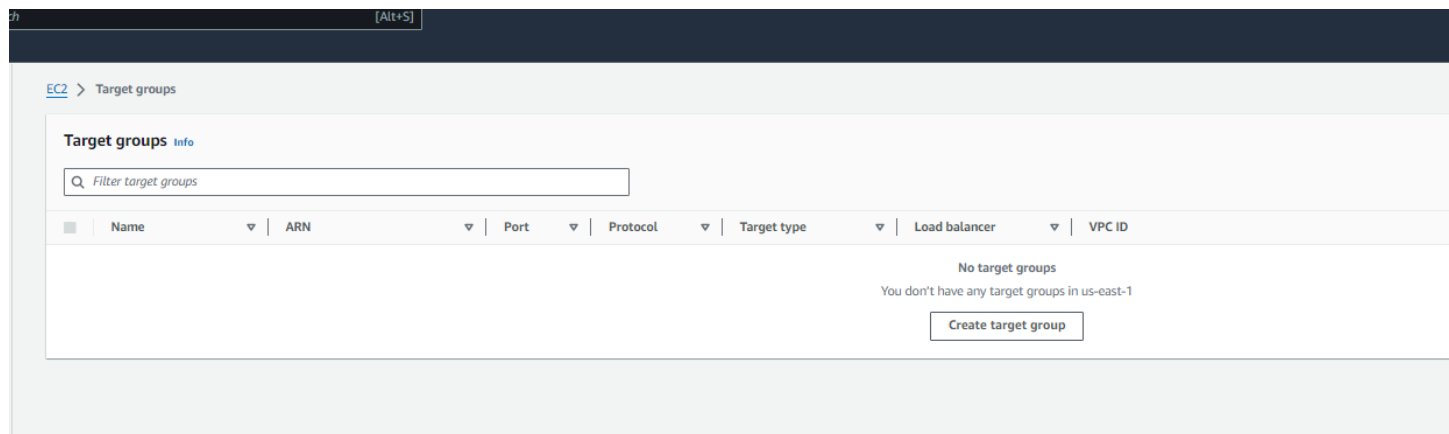
Firewall (security group)
launch-wizard-2

Storage (volumes)
1 volume(s) - 8 GB

Cancel Create Launch Template

2) Create Target Group

Step I : Go to Ec2, In Load Balancing click on target Group then click on create target group.



Step II : Give the name to Target group give necessary information and click on next.

Basic configuration
Settings in this section can't be changed after the target group is created.

Choose a target type

☒ **Instances**

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

☐ **IP addresses**

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

☐ **Lambda function**

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

☐ **Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

astargetgroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

☒ **IPv4**

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ **IPv6**

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

-
vpc-0ee2a53915c3112cf
IPv4: 172.31.0.0/16

Protocol version

☒ HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP

Health check path
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.
/
Up to 1024 characters allowed.

► **Advanced health check settings**

Attributes

ⓘ Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► **Tags - optional**
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

Step III : Leave all as default and click on create target group.

roup

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (0)

Filter instances

Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
No instances							

0 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

Include as pending below

Review targets

Targets (0)

Filter targets ☐ Show only pending

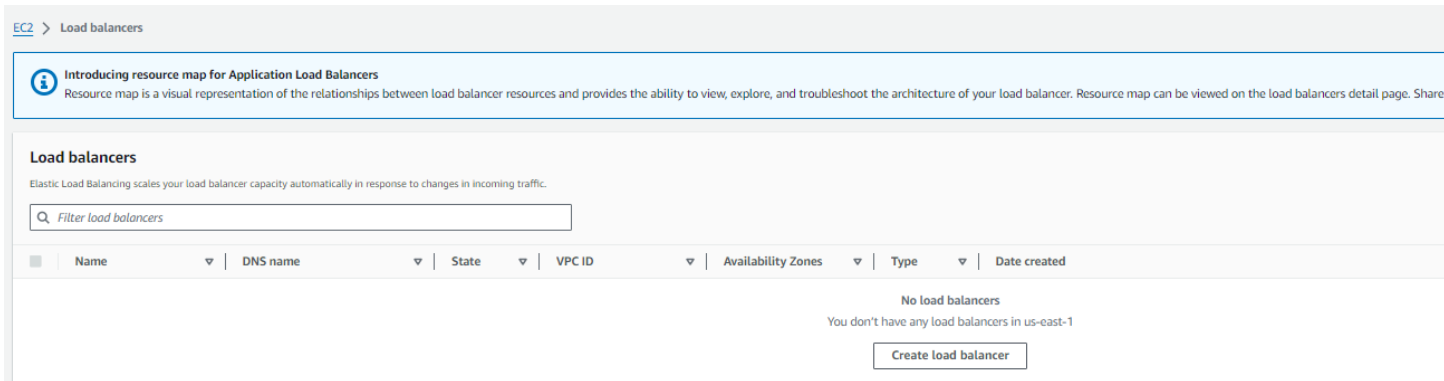
Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
No instances added yet Specify instances above, or leave the group empty if you prefer to add targets later.								

0 pending

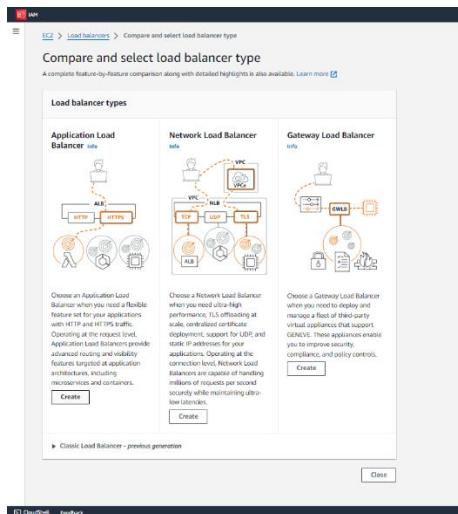
Cancel Previous **Create target group**

3) Create Load Balancer.

Step I : Go to EC2 ,In Load Balancing click on create Load Balancer.



Step II : Select The application load Balancer.



Step III : Give the name to Load Balancer, give mapping, port number and target group.

Create Application Load Balancer [info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

jsloadbalancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [info](#)

Scheme can't be changed after the load balancer is created.

☒ Internet-facing

An Internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [info](#)

Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-

vpc-0e02a20915c1112cf

IPv4: 172.31.0.0/16

Mappings [info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1

vpc-0e02a20915c1112cf
IPv4: 172.31.0.0/16

Mappings [info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ us-east-1a (use1-az2)

Subnet
subnet-09d865a39c0fa9-5c
IPv4 address
Assigned by AWS

☒ us-east-1b (use1-az4)

Subnet
subnet-011b0dea2c5808a49
IPv4 address
Assigned by AWS

☒ us-east-1c (use1-az6)

Subnet
subnet-07912fcb02c803f4
IPv4 address
Assigned by AWS

☒ us-east-1d (use1-az1)

Subnet
subnet-0b449152f57d61844
IPv4 address
Assigned by AWS

☒ us-east-1e (use1-az3)

Subnet
subnet-02311cab18650214
IPv4 address
Assigned by AWS

☒ us-east-1f (use1-az5)

Subnet
subnet-066546a50f5789ea
IPv4 address
Assigned by AWS

CloudShell

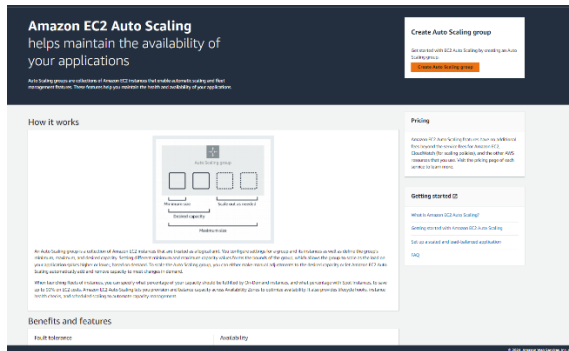
Feedback

[CloudShell](#) [Feedback](#)

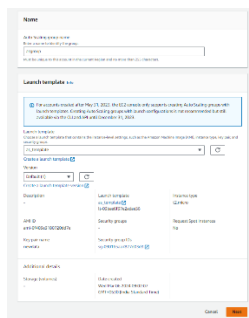
[Help](#) [Feedback](#)

4) Create auto scaling group

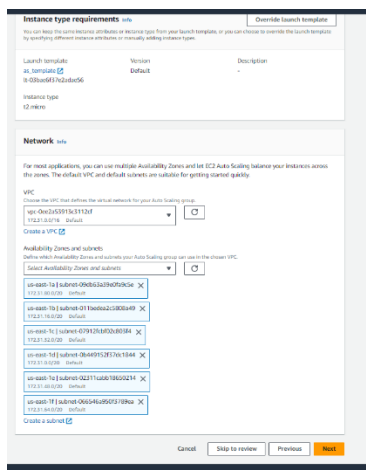
Step I : In Ec2 ,Click on auto Scaling and then create auto Scaling group.



Step II : Give the name and select launch template.click on next.



Step III : Choose Availability zones and attach it to the load balancer,and next.



Step IV : Give the desired ,min, max size to instances also define scaling policies.

Step V : then click next.

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▾

Desired capacity

Specify your group size.

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

Equal or less than desired capacity

Max desired capacity

Equal or greater than desired capacity

Automatic scaling - optional [Info](#)

Choose whether to use a target tracking policy

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☒ No scaling policies

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☐ Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

VPC Lattice integration options [Info](#)

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

☒ No VPC Lattice service

VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

[Create new VPC Lattice service](#)

☐ Attach to VPC Lattice service

Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

☒ Always enabled

Additional health check types - optional [Info](#)

☐ Turn on Elastic Load Balancing health checks **Recommended**

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

☐ Turn on VPC Lattice health checks

VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check grace period [Info](#)

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

 seconds

Additional settings

Monitoring [Info](#)

☐ Enable group metrics collection within CloudWatch

Default instance warmup [Info](#)

The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

☐ Enable default instance warmup

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

Step VI : then click next.

Instance maintenance policy - new [Info](#)

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Control availability and cost during replacement events ×

An instance maintenance policy determines how much availability your application has when EC2 Auto Scaling replaces instances. It also establishes guardrails that limit the amount of capacity that can be added or removed when replacing instances.

Choose a replacement behavior depending on your availability requirements

Mixed behavior
☒ **No policy**
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability
☐ **Launch before terminating**
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs
☐ **Terminate and launch**
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

Flexible
☐ **Custom behavior**
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection

Scale-in protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

☐ Enable instance scale-in protection

Cancel

Skip to review

Previous

Next

Create Auto Scaling group

Add notifications - *optional* [Info](#)

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

Cancel

Skip to review

Previous

Next

> Create Auto Scaling group

Add tags - *optional* [Info](#)

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

i You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group. ×

Tags (0)

Add tag

50 remaining

Cancel

Previous

Next

Step VII : then click on create auto scaling group.

Step 7: Review

Scaling

Minimum desired capacity: 5
Maximum desired capacity: 7
Target tracking policy: (None)

Instance maintenance policy

Replacement behavior: No policy
Min healthy percentage: +
Max healthy percentage: +

Instance scale-in protection

Instance scale-in protection: Off (Enable instance protection from scale in)

Step 5: Add notifications [Edit]

Notifications

No notifications

Step 6: Add tags [Edit]

Tags (0)

Key	Value	Tag new instances
No tags		

Cancel Previous **Create Auto Scaling group**

Step VIII : here,auto-scaling group is created.

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	asgroup	as_template Version Default	0	Updating capacity...	5	3	7	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f

Step IX : Instances are also Launched see here ,5 instances are created here as desired capacity, name it .

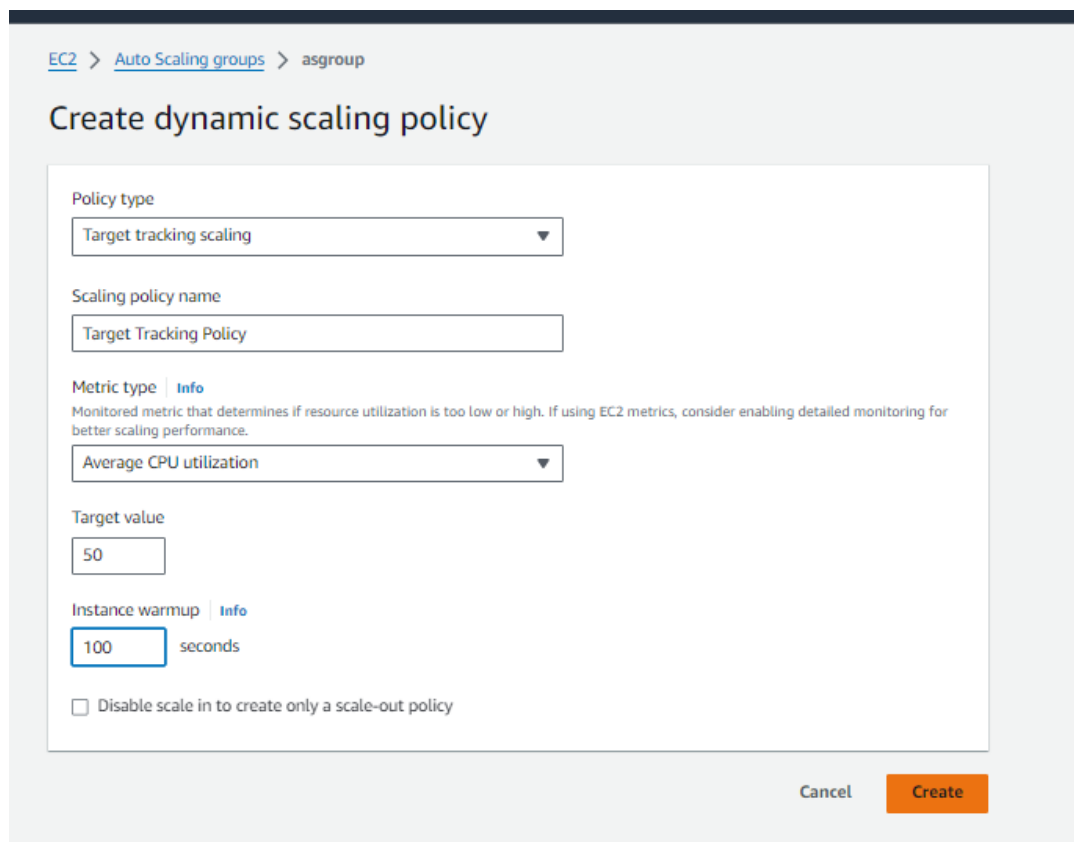
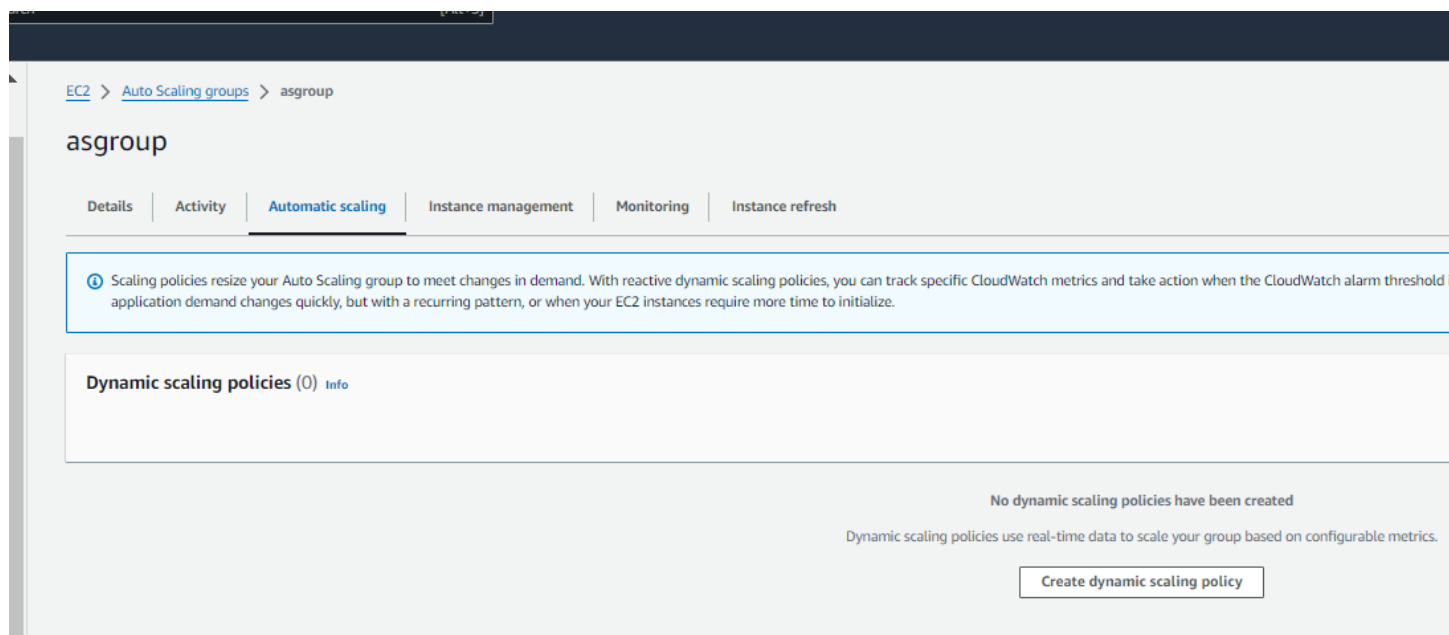
Instances (1/5) Info

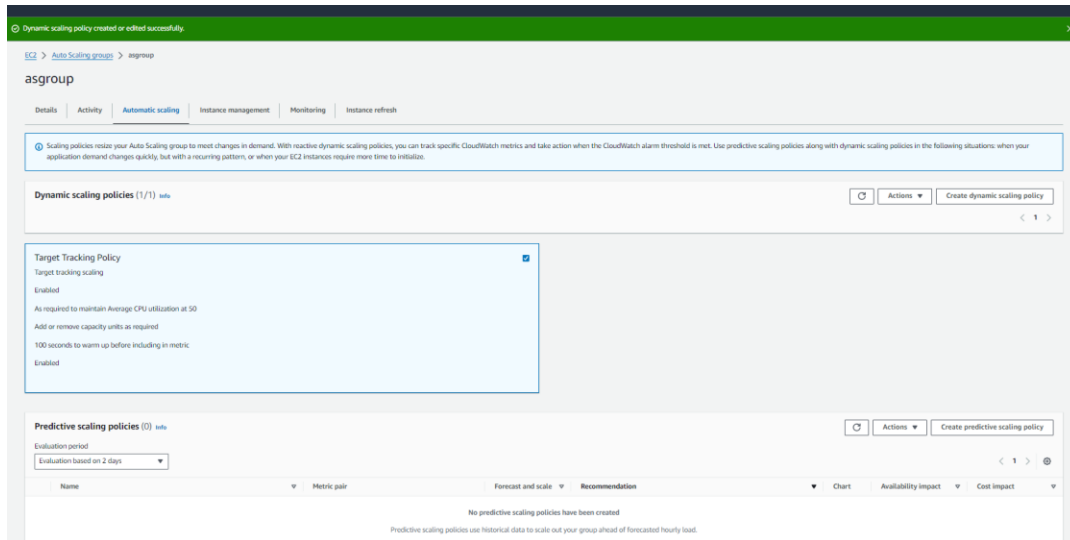
Find Instance by attribute or tag (case-sensitive)

Any state

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zones
<input type="checkbox"/>	1	i-0cf1fede6d670e8dd	Running	t2.micro	Initializing	View alarms +	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f
<input type="checkbox"/>	2	i-0888233d01dbc74b3	Running	t2.micro	Initializing	View alarms +	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f
<input type="checkbox"/>	3	i-0c6e12d69de3448ca	Running	t2.micro	Initializing	View alarms +	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f
<input type="checkbox"/>	4	i-0694e2aafec670c1d	Running	t2.micro	Initializing	View alarms +	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f
<input checked="" type="checkbox"/>	5	i-005b6bda1fc39970e	Running	t2.micro	Initializing	View alarms +	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f

Step X : click on scaling group name and go to Automatic scaling here click on create dynamic scaling policy.





To manage instance:

Step XI : Select any one instance copy its ip go to cmd change the directory where your key is present and then give command

i)ssh -i newdata.pem ec2-user@public_ip_of_instance it is for take remote access.

ii)install the command using “ sudo yum install stress”

iii) after that enter command “stress –help” .


iv) copy these command change cpu 88 and give 15m and paste it for running background give “&” at the end.

v)then give “top” to check Load Average.

Follow this step to all instances to give the stress to instances and increase the load of CPU.

```
ec2-user@ip-172-31-91-112:~$ ssh -i newdata.pem ec2-user@44.203.57.156
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenovo\Downloads>ssh -i newdata.pem ec2-user@44.203.57.156
The authenticity of host '44.203.57.156 (44.203.57.156)' can't be established.
ED25519 key fingerprint is SHA256:1tZTR7L/oLPjX4HLYXgwwLCkKwpWn1KBYYWx0EEy3uM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.203.57.156' (ED25519) to the list of known hosts.
```



```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-91-112 ~]$ sudo yum install stress
Last metadata expiration check: 0:10:37 ago on Wed Mar  6 03:51:46 2024.
Dependencies resolved.

=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
stress                         x86_64            1.0.4-28.amzn2023.0.2   amazonlinux       37 k

Transaction Summary
=====
Install 1 Package

Total download size: 37 k
Installed size: 78 k
Is this ok [y/N]: y
```

```

--backoff N      wait factor of N microseconds before work starts
-c, --cpu N      spawn N workers spinning on sqrt()
-i, --io N       spawn N workers spinning on sync()
-m, --vm N       spawn N workers spinning on malloc()/free()
    --vm-bytes B  malloc B bytes per vm worker (default is 256MB)
    --vm-stride B touch a byte every B bytes (default is 4096)
    --vm-hang N   sleep N secs before free (default none, 0 is inf)
    --vm-keep     redirty memory instead of freeing and reallocating
-d, --hdd N      spawn N workers spinning on write()/unlink()
    --hdd-bytes B write B bytes per hdd worker (default is 1GB)

```

Example: `stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 10s`

```

[ec2-user@ip-172-31-1-253 ~]$ stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[1] 26620
[ec2-user@ip-172-31-1-253 ~]$ stress: info: [26620] dispatching hogs: 87 cpu, 4 io, 2 vm, 0 hdd
stress --cpu 87 --io 4 --vm 2 stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &^C
[ec2-user@ip-172-31-1-253 ~]$ stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[2] 26760
[ec2-user@ip-172-31-1-253 ~]$ stress: info: [26760] dispatching hogs: 87 cpu, 4 io, 2 vm, 0 hdd
stress --cpu 87 --io 4 --vm 2 ^C
[ec2-user@ip-172-31-1-253 ~]$ stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[3] 26856
[ec2-user@ip-172-31-1-253 ~]$ stress: info: [26856] dispatching hogs: 87 cpu, 4 io, 2 vm, 0 hdd

[ec2-user@ip-172-31-1-253 ~]$ top

```

ec2-user@ip-172-31-1-253:~
×
+
▽

```

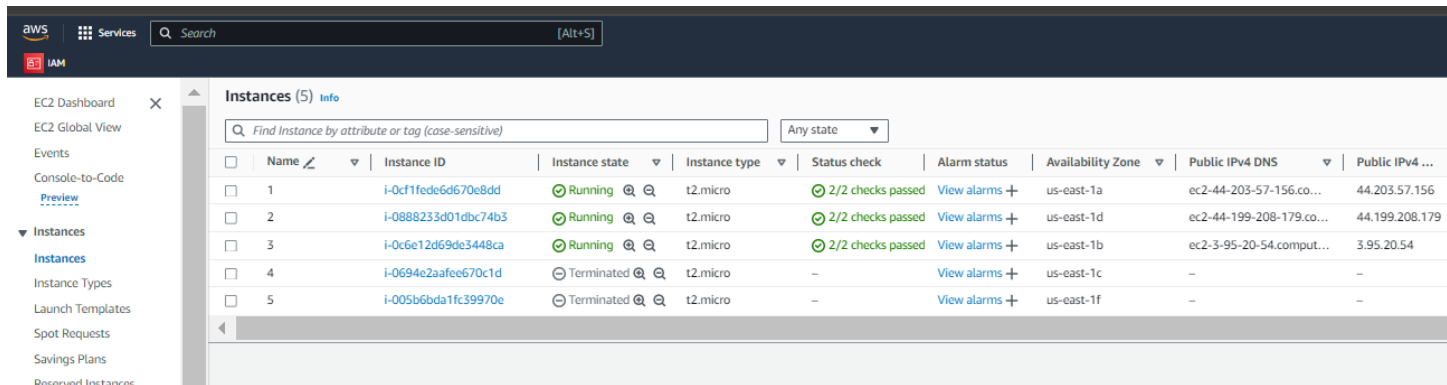
top - 04:11:22 up 20 min,  1 user,  load average: 192.39, 66.97, 24.19
Tasks: 390 total, 270 running, 120 sleeping,   0 stopped,   0 zombie
%Cpu(s): 97.2 us,  2.8 sy,   0.0 ni,   0.0 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
MiB Mem :   949.6 total,    90.9 free,   785.6 used,    73.1 buff/cache
MiB Swap:    0.0 total,    0.0 free,    0.0 used.   56.3 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
26825	ec2-user	20	0	3512	112	0	R	0.7	0.0	0:00.43	stress
26862	ec2-user	20	0	134588	93296	264	R	0.7	9.6	0:00.12	stress
26623	ec2-user	20	0	134588	117584	264	R	0.5	12.1	0:00.86	stress
26624	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26626	ec2-user	20	0	134588	96992	264	R	0.5	10.0	0:00.86	stress
26627	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26629	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.85	stress
26631	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26632	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26636	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26638	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26639	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26640	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26641	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26643	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26644	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26646	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26647	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26649	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26650	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26652	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26653	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress
26654	ec2-user	20	0	3512	108	0	R	0.5	0.0	0:00.86	stress

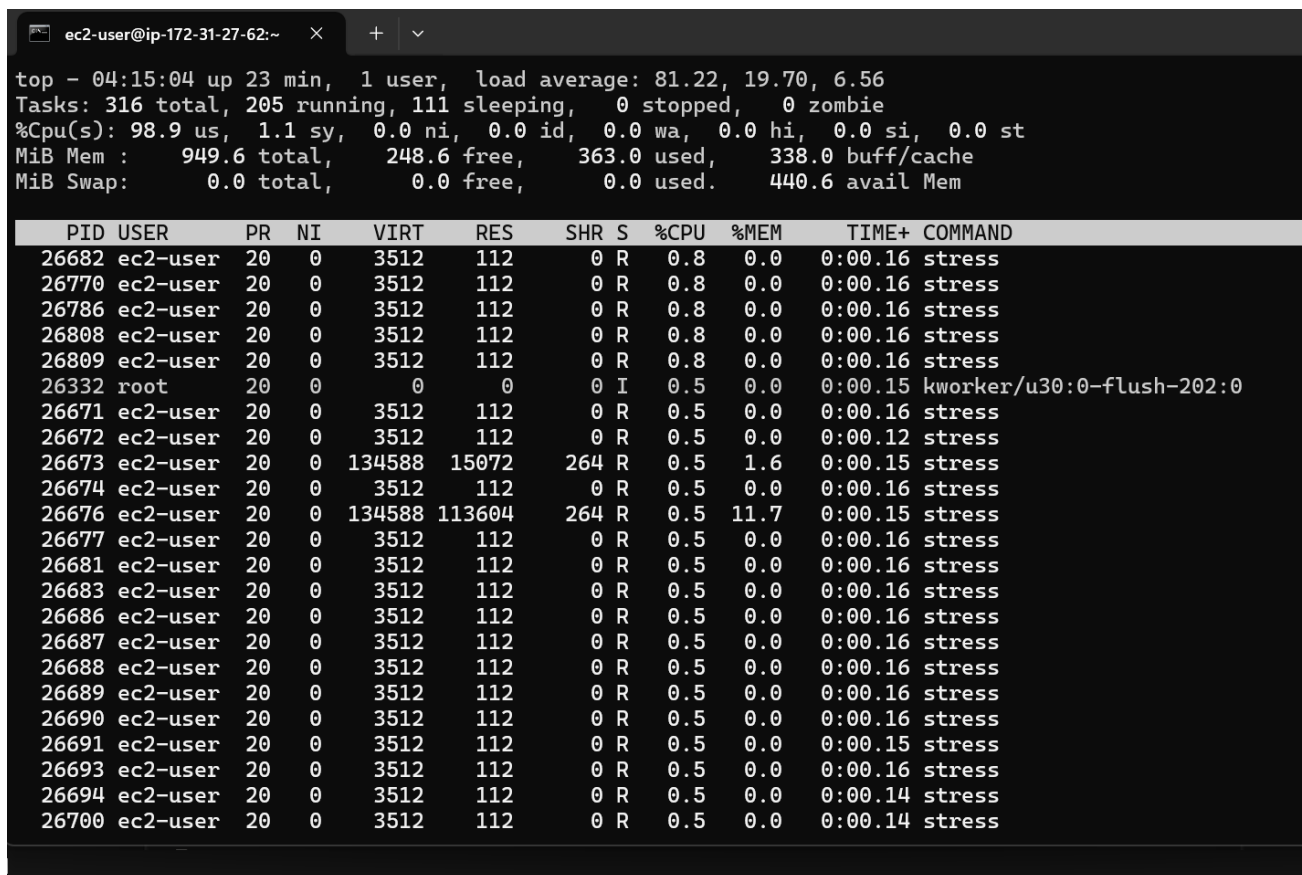
Daily
Alina.pem
2/29/2024 9:50 PM
PEM File

Step XII : Here, load is minimum that why here only minimum capacity of instances running.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	1	i-0cf1fed6d670e8dd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-44-203-57-156.co...	44.203.57.156
<input type="checkbox"/>	2	i-0888233d01dbc74b3	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-199-208-179.co...	44.199.208.179
<input type="checkbox"/>	3	i-0c6e12d69dc3448ca	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-95-20-54.comput...	3.95.20.54
<input type="checkbox"/>	4	i-0694e2aafec670c1d	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-
<input type="checkbox"/>	5	i-005b6bda1fc39970e	Terminated	t2.micro	-	View alarms +	us-east-1f	-	-

Step XIII : we add maximum load on it using above command for all instneces



```
ec2-user@ip-172-31-27-62:~  
top - 04:15:04 up 23 min, 1 user, load average: 81.22, 19.70, 6.56  
Tasks: 316 total, 205 running, 111 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 98.9 us, 1.1 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 949.6 total, 248.6 free, 363.0 used, 338.0 buff/cache  
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 440.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
26682	ec2-user	20	0	3512	112	0	R	0.8	0.0	0:00.16	stress
26770	ec2-user	20	0	3512	112	0	R	0.8	0.0	0:00.16	stress
26786	ec2-user	20	0	3512	112	0	R	0.8	0.0	0:00.16	stress
26808	ec2-user	20	0	3512	112	0	R	0.8	0.0	0:00.16	stress
26809	ec2-user	20	0	3512	112	0	R	0.8	0.0	0:00.16	stress
26332	root	20	0	0	0	0	I	0.5	0.0	0:00.15	kworker/u30:0-flush-202:0
26671	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26672	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.12	stress
26673	ec2-user	20	0	134588	15072	264	R	0.5	1.6	0:00.15	stress
26674	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26676	ec2-user	20	0	134588	113604	264	R	0.5	11.7	0:00.15	stress
26677	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26681	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26683	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26686	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26687	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26688	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26689	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26690	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26691	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.15	stress
26693	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.16	stress
26694	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.14	stress
26700	ec2-user	20	0	3512	112	0	R	0.5	0.0	0:00.14	stress

```
ac2-user@ip-172-31-91-112:~$ top - 04:24:48 up 33 min, 1 user, load average: 1.03, 33.90, 41.22
Tasks: 107 total, 1 running, 106 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 949.6 total, 441.8 free, 169.5 used, 338.2 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 634.3 avail Mem

26426 ec2-user 20 0 3512 108 0 R 1.3 0.0 0:06.82 stress
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slab_flushmq
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
8 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events_highpri
10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_kthread
12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
13 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
14 root 20 0 0 0 0 S 0.0 0.0 0:00.15 ksoftirqd/0
15 root 20 0 0 0 0 I 0.0 0.0 0:00.10 rcu_preempt
16 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
17 root 20 0 0 0 0 I 0.0 0.0 0:00.10 kworker/0:1-cgroup_destroy
18 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
20 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
21 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 inet_frag_wq
22 root 20 0 0 0 0 S 0.0 0.0 0:00.08 kauditd
23 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
24 root 20 0 0 0 0 S 0.0 0.0 0:00.00 nom_reaper
25 root 20 0 0 0 0 I 0.0 0.0 0:09.67 kworker/u30:1-flush-202:0
27 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback
28 root 20 0 0 0 0 S 0.0 0.0 0:00.03 kcompactd0
29 root 39 19 0 0 0 S 0.0 0.0 0:00.00 khugepaged
30 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd
31 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd
32 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 blkcg_punt_bio
33 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xen-balloon
```

Step XIV : here see to load increased that’s why maximum capacity of instances are add to distribute load.

Instances (7) Info										
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> Any state										
<input type="checkbox"/>	Name ↗	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS ▼	Public IPv4 ... ▼	Elastic IP
<input type="checkbox"/>	newadd	i-06ad81f3cba11962f	Running 🔍 🔍	t2.micro	2/2 checks passed View alarms +		us-east-1e	ec2-100-26-217-14.co...	100.26.217.14	–
<input type="checkbox"/>	1	i-0cf1fede6d670e8dd	Running 🔍 🔍	t2.micro	2/2 checks passed View alarms +		us-east-1a	ec2-44-203-57-156.co...	44.203.57.156	–
<input type="checkbox"/>	2	i-0888233d01dbc74b3	Running 🔍 🔍	t2.micro	2/2 checks passed View alarms +		us-east-1d	ec2-44-199-208-179.co...	44.199.208.179	–
<input type="checkbox"/>	newadd2	i-0d98ce9a39b59192a	Running 🔍 🔍	t2.micro	2/2 checks passed View alarms +		us-east-1f	ec2-3-95-27-233.comp...	3.95.27.233	–
<input type="checkbox"/>	3	i-0c6e12d69de3448ca	Running 🔍 🔍	t2.micro	2/2 checks passed View alarms +		us-east-1b	ec2-3-95-20-54.comput...	3.95.20.54	–
<input type="checkbox"/>	4	i-0694e2aafec670c1d	Terminated 🔍 🔍	t2.micro	– View alarms +		us-east-1c	–	–	–
<input type="checkbox"/>	5	i-005b6bda1fc39970e	Terminated 🔍 🔍	t2.micro	– View alarms +		us-east-1f	–	–	–

Step XV : After doing that, kill the stress give command “killall stress”.